8Ш

УДК 004.8

Е.А. Шалфеева
Институт автоматики и процессов управления ДВО РАН, г. Владивосток, Россия
shalf@iacp.dvo.ru

Модель связи структурных свойств онтологии со структурой модели требований системы, основанной на редактируемых знаниях*

Предлагается модель связи онтологии, лежащей в основе разработки системы, с набором требований к ней. Минимальный набор требований формируется на основе онтологии предметной области и с учетом традиционного распределения функций между подсистемами такой системы. Модель обеспечивает возможность автоматического формирования таких требований.

Создание и внедрение экспертных систем выдвинуло проблему технологической поддержки разработок в области искусственного интеллекта на передний план [1]. Ввиду все возрастающего использования систем искусственного интеллекта в конкретных приложениях, к ним начинают предъявляться практически те же требования, что и к традиционным программным комплексам и системам. В связи с этим становится весьма актуальной поддержка жизненного цикла разработки этих систем. Как и в случае обычных программных систем, разработка системы искусственного интеллекта должна начинаться с формулирования *полных, непротиворечивых и однозначных требований* к ней [2]. Таким образом, создание ПО систем, основанных на знаниях, (СОЗ) имеет как общие моменты с разработкой традиционных систем ПО, так и свою специфику, которая явным образом должна отражаться в соответствующих моделях жизненного цикла.

Традиционно при разработке СОЗ используется формализованная онтология предметной области, а сами СОЗ включают в себя типичные архитектурные компоненты, в числе которых есть *редактор знаний* и *подсистема решения* задач пользователя. *Редактор знаний* вместе с базой знаний являются относительно независимой подсистемой, ориентированной на своего пользователя – эксперта предметной области и инженера знаний. Поэтому функциональные требования к ней могут рассматриваться отдельно от требований к остальной части СОЗ, ориентированной на специалиста, решающего задачи в своей предметной области.

В статье рассматривается возможность поддержки автоматического формирования требований к обеим подсистемам на основе онтологий, лежащих в основе разработки системы.

Постановка задачи

Формулирование полных, непротиворечивых и однозначных требований начинается с их явного представления в виде документа. Известно, что учет разных точек зрения при создании требований обеспечивает их максимальную адекватность. Важны как точки зрения разных участников разработки, так и разные аспекты их представления – функциональный, информационный, поведенческий.

* Работа выполнена при финансовой поддержке ДВО РАН, проект «Обеспечение качества интеллектуального программного обеспечения в процессе его проектирования на основе онтологий».

Онтология предметной области, лежащая в основе разрабатываемой СОЗ, содержит согласованное мнение экспертов, а с другой стороны, охватывает функциональный и информационный аспекты анализа требований к программному обеспечению, разрабатываемому в предметной области.

В *модели онтологии предметной области* целесообразно различать две составляющие – *систему понятий знаний* (вместе с ограничениями целостности знаний) и *систему понятий действительности* (вместе с ограничениями целостности ситуаций).

Обе составляющие по форме представления схожи и соответствуют типичным представлениям об онтологии, как взаимосвязанной совокупности определений терминов-сущностей, терминов-связей и онтологических соглашениях об ограничениях на их интерпретацию:

$$Онтология = (\{Сущность_k\},\{Связь_l\},\{Соглашение_m\}).$$

Часто формальное представление *системы понятий знаний* и *системы понятий действительности* размещены в разных теориях (модулях) онтологии предметной области. Например, для решения задачи построения рентгеновского спектра терминами *системы понятий действительности* являются: *источник возбуждающего излучения, активность источника, относительная эффективность регистрации, энергетическое разрешение, угол падения первичного излучения, угол отбора вторичного излучения, количество каналов, коэффициенты калибровки, плотность пробы, толщина пробы* и др. А *система понятий знаний* включает: *характеристические линии элемента, энергия характеристического излучения, энергии линий изотопа, сечение возбуждения* и др.

Выделяют также и такую часть модели онтологии, как *связь знаний и действительности*, которая, например, может содержать все формулы, требующиеся для получения решения конкретных задач.

Такое разделение позволяет использовать *систему понятий знаний* (вместе с ограничениями целостности знаний) для поддержки формирования требований к редактору знаний, а *систему понятий действительности* и *связь знаний и действительности* – для поддержки формирования требований к подсистеме решения задач. Требуется получить модель связи онтологии с набором требований, которые могут быть получены исходя из описания предметной области и из традиционного распределения функций между подсистемами СОЗ. Прочие требования, если они есть, будут добавлены аналитиком к этому методологически обоснованному набору требований.

Формальное представление онтологии дает возможность извлекать ее внутреннее содержание, структуру и оценивать внутренние свойства, это позволяет построить модель связи структурных свойств онтологии со структурой модели требований. Получаемый таким способом набор требований должен обеспечить «каркас» для их точной спецификации.

Формирование требований к СОЗ

Традиционно документ, описывающий требования к программному обеспечению, включает в себя такие подразделы (группы требований), как: общие функциональные требования, требования к качеству и эксплуатационным характеристикам, описание информации, описание поведения, спецификации функций, описание критериев для подтверждения и др.

Общие функциональные требования включают описание предназначения всей СОЗ, их написание не является рутинной работой и не требует автоматизации. Пример формулировки для СОЗ, выполняющей построение характеристического рентгеновского спектра[1], таков. «Программная система должна выполнять построение ха-

[1] Первая версия разработана в лаборатории интеллектуальных систем ИАПУ ДВО РАН.

рактеристического рентгеновского спектра по известным элементному составу пробы и ее геометрическим свойствам, позволять выполнять перемещение по графику и его масштабирование, позволять сохранять полученный спектр в файл, позволять добавлять, изменять и удалять знания, используемые при построении спектра».

Далее для СОЗ явно указывается (или оно очевидно) распределение общих функций по подсистемам, как минимум, по двум – подсистемы для эксперта и для пользователя. Например, редактор знаний должен: позволять пользователю просматривать, добавлять, изменять и удалять информацию обо всех сущностях, знания о которых не зависят от текущего состояния действительности, но используются для решения задач. «Построитель» должен: позволять пользователю задавать входные условия задачи, решать задачу и обеспечивать удобство при просмотре результатов.

Более конкретные функциональные требования (типичный минимум для таких систем) и описание информации создаются на основе метода анализа структуры онтологий [2], [3]. Требования к качеству и эксплуатационным характеристикам слишком мало связаны с моделью предметной области, поэтому к рассматриваемой методологии не относятся. Описание поведения мало связано с моделью предметной области, но может быть формализовано на основе модели пользовательских задач, которая базируется на онтологии пользовательских задач. Формализация описания поведения возможна, но основана на другой онтологии и модели. А описание критериев для подтверждения достаточно легко формулируется при наличии готовых спецификаций функций и описании используемой информации.

Модель для формирования требований к подсистеме редактирования знаний. Описание функций

Наиболее популярный случай определения сущностей в онтологии знаний – их описание названием и атрибутами:

Сущность = (Название, 1..{Атрибут*$_i$*}).*

Структура соответствующего фрагмента текста онтологии знаний (на языке прикладной логики [4]) такова:

сорт «названиеСущ»: {}n;

{сорт «названиеАтр»: («названиеСущ» → СтдТип)},

где СтдТип – предопределенное множество значений или его подмножество, например, все положительные целые.

Кроме формального подхода к выявлению атрибутов сущности может быть применен анализ смысла названия, а именно: «названиеАтр», как правило, не указывает на процесс или действие.

Менее распространены случаи, когда сущность задается только своим названием, либо задается как частный случай другой сущности (или подмножество некоторого множества).

Типичный набор функций для редактора знаний связан с вводом всех таких сущностей (составных объектов) и их атрибутов: добавление сущности, добавление \ изменение одного или нескольких атрибутов, удаление сущности.

Модель отображения структуры онтологии на структуру требований включает следующую пару (элементы онтологии – элементы требований).

Онтология. Сущность = (Название, 1..{Атрибут*$_i$*})*

Требования.

ФункцииРедактированиеДанных.

Ф-Добавить-Экземпляр-Сущности<Название>;

Вход: Название,
Выход: обновленная таблица экземпляров сущности;
Ф-Удалить-Экземпляр-Сущности<Название>;
Вход: Название,
Выход: обновленная таблица экземпляров сущности;
1..{Ф-Изменить-Атрибут<Атрибут$_i$>Экземпляра-Сущности<Название>*
Вход: Атрибут$_i$,
Выход: обновленный набор атрибутов экземпляра сущности;
}

Достаточно распространено в онтологии знаний задание агрегирующих связей, когда сущность включает в себя набор однотипных частей либо характеризующих ее других сущностей:

Онтология.Связь = (ХарактеристикаИлиЧасть,
НазваниеСущности,
1..{ОблЗначХарактеристикаИлиЧасть$_i$}),*

здесь 1..* – указывает на кардинальность связи *сущности* со своей *характеристикой или частью* – «не менее одного».

В этом случае к набору функций редактора знаний должен быть добавлен ввод всех этих компонентов сущностей.

Обычно в онтологии задаются также свойства этих компонентов (*характеристик или частей*) – так называемые совместные свойства сущности и ее компонента.

Онтология. Связь = (НазваниеСвязи, (НазваниеСущности,
ОблЗначХарактеристикаИлиЧасть), ОблЗначСвязи)

Это требует добавления к функциональности редактора *ввода всех «совместных свойств» (добавить запись о совместных свойствах, изменить одно или несколько совместных свойств, удалить запись)*.

Пример. В онтологии знаний вышеупомянутой задачи каждому элементу сопоставлены «энергетические уровни»:

<u>*сорт*</u> *энергетические уровни элемента:*
(химические элементы $\rightarrow$ {}энергетические уровни).

Есть также функции, сопоставляющие всем таким парам некоторые действительные числа:

<u>*сорт*</u> *энергия связи: ({(v: (× химические элементы, энергетические уровни))*
$\pi(2, v) \in$ энергетические уровни элемента($\pi(1, v)$)} $\rightarrow r(0; +\infty)$)
<u>*сорт*</u> *скачок поглощения: ({(v: (× химические элементы, энергетические уровни))*
$\pi(2, v) \in$ энергетические уровни элемента($\pi(1, v)$)} $\rightarrow r(1; +\infty)$)

Модель отображения структуры онтологии на структуру требований включает также следующие блоки (вход и выход функций для краткости опустим).

Онтология. Связь =(ХарактеристикаИлиЧасть, НазваниеСущности,
1..{ОблЗначХарактеристикаИлиЧасть$_i$})*
Требования.
ФункцииРедактированиеДанных.
Ф-Добавить-Экземпляр-ХарактеристикиИлиЧасти
<ХарактеристикаИлиЧасть>Экземпляра-Сущности
<НазваниеСущности>;
Ф-Удалить-Экземпляр-ХарактеристикиИлиЧасти
<ХарактеристикаИлиЧасть>Экземпляра-Сущности
<НазваниеСущности>;

и

Онтология. Связь = (НазваниеСвязи, (НазваниеСущности,
ОблЗначХарактеристикаИлиЧасть), ОблЗначСвязи)

И
Онтология. Связь = (НазваниеСущности, 1..{ХарактеристикаИлиЧасть $_i$})*
Требования.
ФункцииРедактированиеДанных.
Ф-Добавить-СовмСвойство<НазваниеСвязи>Сущности <НазваниеСущности> и Сущности<ХарактеристикаИлиЧасть>;
Ф-Удалить-СовмСвойство<НазваниеСвязи>Сущности< НазваниеСущности> и Сущности<ХарактеристикаИлиЧасть>;
1..{Ф-Изменить-Значение-Свойства<НазваниеСвязи>Сущности< НазваниеСущности>и Сущности< ХарактеристикаИлиЧасть>;*

Ограничения на значения сущностей и/или их свойств являются важной информацией при разработке редактора знаний. Онтология знаний и ее раздел «Ограничения целостности знаний» позволяет получить эту информацию.

Каждое онтологическое соглашение является источником функционального требования к редактору знаний, если оно содержит в качестве операндов вводимые и изменяемые сущности, атрибуты, характеристики, части, свойства.

Например, предложение онтологии

(v: энергетические уровни) орбитальное квантовое число(v) < главное квантовое число энергетического уровня(v)

приводит к необходимости функции контроля вводимой информации в соответствии с утверждением: при вводе орбитального квантового числа любого энергетического уровня электронной оболочки проверить, чтобы оно было меньше главного квантового числа электронной оболочки. Такая проверка должна быть активизирована в редакторе всякий раз, когда вводится новое значение атрибута – орбитальное квантовое число и главное квантовое число, чтобы сопоставить с введенными ранее значениями.

Для анализа очередности активизации функции (чтобы работала в паре с соответствующей функцией ввода) важно увидеть в структуре соглашения отдельные операнды, а для формулирования функции важен сам текст. Поэтому минимальная модель структуры соглашения такова.

*Онтология.*Соглашение = *Онтология.*Соглашение.*Текст*
+ *Онтология.*Соглашение.*Операнды*
*Онтология.*Соглашение.*Операнды = {ЗначениеСвойства| ЗначениеАтрибута| ЗначениеХарактеристикиИлиЧасти}*

Если есть в соглашениях условия на некоторое данное, то в редакторе при ввод/добавлении/изменении должны быть подфункции, обеспечивающие выполнение этих условий. Для каждого отдельно добавляемого данного – своя функция проверки или набор подфункций по каждому условию.

Модель отображения структуры онтологии на структуру требований должна включать следующий блок.

*Онтология.*Соглашение = *Онтология.*Соглашение.*Текст* +
*Онтология.*Соглашение.*Операнды*
И
Операнд$_j$= Атрибут$_i$ | ЗначениеХарактеристикиИлиЧасти$_s$
Требования.
ФункцииПроверкиДанных.
Ф-проверить-выполнение-ограничения.
Вход: Операнд$_j$,
Описание: <Текст>;
Выход: Boolean и сообщение пользователю о некорректности ввода.

Вышеперечисленный (и зафиксированный в модели отображения структуры онтологии на структуру требований) набор требований является проверяемым и однозначно по-

нимаемым. Такой набор может быть передан аналитиком проектировщику или кодировщику, он же становится основой для формулирования критериев для подтверждения. Этими достоинствами не обладает следующая формулировка требования к редактору, созданная вне рассматриваемой методологии: *«При нажатии кнопки "Изменить" рядом с таблицей "Энергетические уровни электронной оболочки" новые значения (в полях ввода) свойств выделенного энергетического уровня выделенной электронной оболочки должны быть проверены на корректность с помощью условий, таких как "У каждого энергетического уровня орбитальное квантовое число меньше главного квантового числа электронной оболочки, которой этот энергетический уровень принадлежит". Если значения корректны, то в базе знаний и в таблице "Энергетические уровни электронной оболочки" значения свойств выделенного энергетического уровня выделенной электронной оболочки должны быть изменены на указанные. Если значения некорректны, то должно быть выведено сообщение о недопустимости указанных значений».* Такая формулировка оправданна, когда аналитик является и проектировщиком, и кодировщиком, и испытателем, т.е. фиксирует требования «для себя».

Модель для формирования требований к подсистеме редактирования знаний. Описание информации

Если программное обеспечение создается в рамках одной предметной области, то ожидается, что все объекты данных редактора знаний присутствуют в онтологии знаний. Анализ теоретико-множественных связей между сущностями в онтологии знаний (и/или таксономических связей) *редактора знаний* позволяет сформировать описание входных данных для редактора знаний.

Модель отображения структуры онтологии на структуру требований включает следующую пару (элементы онтологии – элементы требований).

Онтология. Атрибут = (НазваниеСущности, Название Атрибута, ОбластьЗначений)
Требования.
Спецификация входных данных.
Сущность. Название Атрибута. ОбластьЗначений

Например, на основе фрагмента онтологии знаний

сорт орбитальное квантовое число: (энергетические уровни $\rightarrow i[0; +\infty)$ *)*
и
сорт спин-орбитальное связывание: (энергетические уровни $\rightarrow r[1/2; +\infty)$ *)*

будут сформированы области определений двух атрибутов сущности «энергетические уровни»:

энергетические уровни. орбитальное квантовое число
имеет область значений целые числа в диапазоне $[0; +\infty)$*;*
энергетические уровни. спин-орбитальное связывание
имеет область значений действительные числа в диапазоне $[1/2; +\infty)$ *).*

Модель для формирования требований к подсистеме решения задач

Терминологию, связанную с решением задач, отражают такие части онтологии, как *система понятий действительности* и *связь знаний и действительности.* Кроме того, необходима спецификация задач или класса задач для разрабатываемой СОЗ, которая позволяет выделить среди терминов *система понятий действительности* те, что являются пользовательскими входными данными, и те – что выходными.

Для подсистемы решения задач характерны такие группы функций, как функции «*вычислительные» и функции «взаимодействия с пользователем».*

Формирование «вычислительных» функций по онтологии включает два этапа. Первый на основе *системы понятий действительности* дает спецификацию названия и входа-выхода функции.

Например, по предложениям онтологии:

<u>сорт</u> элементы пробы: {}химические элементы
характеристические линии ≡ радиационные переходы
<u>сорт</u> радиационные переходы: {}n
<u>сорт</u> интенсивность характеристической линии: ({(v: (× элементы пробы, характеристические линии)) π(2, v) ∈ характеристические линии элемента(π(1, v))} → r[0; +∞))
<u>сорт</u> площадь пика: ({(v: {(w: (× элементы пробы, характеристические линии)) π(2, w) ∈ характеристические линии элемента(π(1, w)}) интенсивность характеристической линии(v) > 0} → r(0; +∞))

формируются первичные спецификации двух функций:

Найти интенсивность характеристической линии
Вход: 1) элементы пробы: множество подмножеств множества «химические элементы»,
2) характеристические линии: множество подмножеств множества названий
Выход: *интенсивность:* Действительное число в диапазоне [0; +∞)

Найти площадь пика
Вход: 1) элементы пробы: множество подмножеств множества «химические элементы»,
2) характеристические линии: множество подмножеств множества названий
Выход: *площадь пика:* Действительное число в диапазоне [0; +∞)

Модель отображения структуры онтологии действительности на структуру требований включает следующую пару (элементы онтологии – элементы требований).

Онтология. Связь = (НазваниеФункции, 1..{НазваниеАргумента$_j$} Область ЗначенийФункции)*
И
Онтология. Сущность = (НазваниеАргумента$_j$, Область Определения)
Требования.
ФункцииВычисления.
Ф-<НазваниеФункции>.
Вход: 1..{(НазваниеАргумента$_j$, Область Определения)}*
Выход: (НазваниеФункции, Область ЗначенийФункции)

На основе модели «*связь знаний и действительности» спецификация дополняется описанием сути преобразования входных данных в выходные.*

В соответствии с фрагментами вида

Онтология. Формула = ({Локальная Переменная$_j$, ОбластьОпредПеременной$_j$}, НазваниеФункции, ТекстФормулы)

модель отображения расширяется таким образом:

Ф-<НазваниеФункции>.
Вход: 1..{(НазваниеАргумента$_j$, Область Определения)}*
Описание: ТекстФормулы
Выход: (НазваниеФункции, Область ЗначенийФункции)

Пример. *Система понятий действительности содержит предложение:*
(el: элементы пробы) (hl: характеристические линии элемента(el)) **интенсивность характеристической линии(el, hl)** = содержание элемента в пробе(el) * инструментальная постоянная * относительная эффективность регистрации (энергия характеристического излучения(el, hl)) * активность источника * (∑(e: энергии линий изотопа (источник возбуждающего излучения)) выход линии (источник возбуждающего излучения, e) * ((сечение возбуждения(el, hl))(e) * (фактор поглощения(el, hl))(e) + (фактор вторичного возбуждения(el, hl))(e)))

Итоговое требование таково:

Найти интенсивность характеристической линии

Вход: 1) элементы пробы: множество подмножеств множества «химические элементы»,

2) характеристические линии: множество подмножеств множества названий

Описание: (el: элементы пробы) (hl: характеристические линии элемента(el)) **интенсивность характеристической линии(el, hl)** = содержание элемента в пробе(el) * инструментальная постоянная * относительная эффективность регистрации (энергия характеристического излучения(el, hl)) * активность источника * (∑(e: энергии линий изотопа (источник возбуждающего излучения)) выход линии (источник возбуждающего излучения, e) * ((сечение возбуждения(el, hl))(e) * (фактор поглощения(el, hl))(e) + (фактор вторичного возбуждения(el, hl))(e)))

Выход: *интенсивность:* Действительное число в диапазоне [0; +∞)

Вместе с вычислительными функциями важно специфицировать и функции проверки вычисляемых значений, если в онтологии действительности присутствуют соответствующие онтологические соглашения. Модель отображения такая же, как и для редактора знаний.

Пример специфицированной функции:

Проверить ограничение 8

Вход: элементы пробы, характеристические линии, используемые как одна пара (т.е. второй элемент пары вычислен как функция характеристические линии элемента от первого элемента пары)

Описание: Проверить выполнение равенства: (el: элементы пробы) (hl: {(v: характеристические линии элемента(el)) интенсивность характеристической линии(el, v) > 0}) площадь пика(el, hl) = интенсивность характеристической линии(el, hl)

Выход: boolean

Описание интерфейсов с пользователем

Для взаимодействия конечного пользователя с системой разрабатывается графический пользовательский интерфейс, требования к которому включают описание сценария решения каждой задачи, а для каждого шага сценария – функции ввода данных и управляющих воздействий пользователя плюс функции отображения результатов.

Функции ввода и проверки данных от пользователя формируются по модели, аналогичной модели отображения в требования редактора знаний.

Пример:

Ф-Проверить ограничение-на элементы-пробы
Вход: элементы пробы.
Описание: Проверить выполнение элементы пробы ≠ ∅
Выход: boolean

Остальные требования формируются с учетом *модели задач пользователя*. Каждая задача в этой модели разбивается на линейную последовательность шагов[1] и возможное множество альтернативных подпоследовательностей, каждый из «последних» также разбивается на линейную последовательность шагов и возможное множество альтернативных подпоследовательностей. Шаг здесь – некоторое воздействие пользователя через пользовательский интерфейс. В этой модели одинаковые шаги могут повторяться в одной или нескольких задачах, они должны иметь одинаковые названия.

[1] Другой вариант *модели задач пользователя* – граф сценария или карта диалога, еще один вариант – блок-схема сценария.

Пример фрагмента модели задачи пользователя (здесь – эксперта):

Вводит некорректные значения свойств новой электронной оболочки и нажимает кнопку «Добавить»
ИЛИ
Вводит корректные значения свойств новой электронной оболочки и нажимает кнопку «Добавить»
ИЛИ
Выделяет в таблице «Электронные оболочки» какую-то электронную оболочку
и нажимает кнопку «Удалить»
ИЛИ
вводит некорректные новые значения свойств выделенной электронной оболочки и нажимает кнопку «Изменить» рядом с данной таблицей
ИЛИ
вводит корректные новые значения свойств выделенной электронной оболочки и нажимает кнопку «Изменить»
ИЛИ
вводит некорректные значения свойств нового энергетического уровня выделенной электронной оболочки и нажимает кнопку «Добавить»
ИЛИ
вводит корректные значения свойств нового энергетического уровня выделенной электронной оболочки и нажимает кнопку «Добавить»
ИЛИ
Выделяет в таблице «Электронные оболочки» какую-то электронную оболочку, выделяет в таблице «Энергетические уровни электронной оболочки» какой-то энергетический уровень
и нажимает кнопку «Удалить».
ИЛИ
вводит некорректные новые значения свойств выделенного энергетического уровня и нажимает кнопку «Изменить» рядом с последней таблицей
ИЛИ
вводит корректные новые значения свойств выделенного энергетического уровня и нажимает кнопку «Изменить»

Каждый шаг этой модели отображается на две функции требований: одна требует от подсистемы обеспечить пользователю очередное состояние интерфейса, другая – обработать воздействие пользователя (шаг).

Например, в последовательности шагов есть шаг:
Добавить новое свойство выделенной электронной оболочки
Получим две функции:

1. *Выделить выбранную пользователем строку электронной оболочки в таблице* «Электронные оболочки», предоставить *поля ввода значений свойств, сделать доступной кнопку ввода измененных значений.*
2. *Обработать воздействие пользователя:*
   2-1) *проверить новые значения (в полях ввода) свойств выделенной электронной оболочки,*
   2-2) *изменить в базе знаний значения свойств выделенной электронной оболочки должны быть на указанные,*
   2-3) *вывести сообщение о недопустимости указанных значений.*

Введение и рассмотрение онтологии моделей задач позволит построить модель отображения фрагментов модели на требования.

Заключение

Разработка систем, основанных на знаниях, производимая в рамках архитектуры «редактор знаний – решатель – интерфейс – система объяснения», производится на основе технологии, в которой уделено много внимания начальным этапам жизненного цикла, и недостаточно внимания – этапам, традиционным для разработки программного обеспечения. Однако полнота и однозначность описания требований по-прежнему являются «залогом» качества программного обеспечения, в частности СОЗ. Предлагаемая методология, позволяющая автоматизировать поддержку создания требований, добавляет к технологии СОЗ «недостающее звено». В основе этой методологии лежит модель связи структуры онтологии знаний предметной области и действительности с требованиями к создаваемым системам, решающим задачи на основе этих знаний.

Литература

1. Гаврилова Т.А. Базы знаний интеллектуальных систем: учебник / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Изд-во «Питер», 2000. – 382 с.
2. Ontologies for Software Engineering and Software Technology / [Calero C., Ruiz F., Piattini M. (Eds.)]. – Berlin : Springer-Verlag, 2006. – 339 p.
3. Клещев А.С. Определение структурных свойств онтологий / А.С. Клещев, Е.А. Шалфеева // Изв. РАН. Теория и системы управления. – 2008. – № 2. – С. 69-78.
4. Клещев А.С. Необогащенные системы логических соотношений / А.С. Клещев, И.Л. Артемьева // НТИ. Сер. 2. – 2000. – Часть 1. – № 7; Часть 2. – № 8.

О.А. Шалфеєва

Модель зв'язку структурних властивостей онтології зі структурою моделі вимог системи, що заснована на редагованих знаннях

Пропонується модель зв'язку онтології, що лежить в основі розробки системи, з набором вимог до неї. Мінімальний набір вимог формується на основі онтології предметної області та з урахуванням традиційного розподілу функцій між підсистемами такої системи. Модель забезпечує можливість автоматичного формування таких вимог.

Ye.A. Shalfeyeva

The Model of Linkage between Structural Properties Ontology and Requirements Model Structure System Based on Knowledge's Being Edidet

The model of linkage which is at the heart of a system development and system requirements set is offered. Minimal requirements set is formulated on the basis of domain and and taking into account traditional functions distributing between subsystems. The model gives an opportunity for automatic requirements formulation.

Статья поступила в редакцию 11.06.2009.